# A finite volume scheme with preconditioned Lanczos method for two-dimensional space-fractional reaction-diffusion equations

Q. Yang[1,*], I. Turner, T. Moroney, F. Liu

*School of Mathematical Sciences, Queensland University of Technology, GPO Box 2434, Brisbane, QLD 4001, Australia.*

**Abstract**

Fractional differential equations have been increasingly used as a powerful tool to model the non-locality and spatial heterogeneity inherent in many real-world problems. However, a constant challenge faced by researchers in this area is the high computational expense of obtaining numerical solutions of these fractional models, owing to the non-local nature of fractional derivatives. In this paper, we introduce a finite volume scheme with preconditioned Lanczos method as an attractive and high-efficiency approach for solving two-dimensional space-fractional reaction-diffusion equations. The computational heart of this approach is the efficient computation of a matrix-function-vector product $f(\mathbf{A})\mathbf{b}$, where $\mathbf{A}$ is the matrix representation of the Laplacian obtained from the finite volume method and is non-symmetric. A key aspect of our proposed approach is that the popular Lanczos method for symmetric matrices is applied to this non-symmetric problem, after a suitable transformation. Furthermore, the convergence of the Lanczos method is greatly improved by incorporating a preconditioner. Our approach is showcased by solving the fractional Fisher equation including a validation of the solution and an analysis of the behaviour of the model.

*Keywords:* fractional Laplacian, matrix transfer technique, matrix function, Lanczos method, Krylov subspace, preconditioner, finite volume method, fractional reaction-diffusion equations

*Corresponding author

*Email address:* q.yang@qut.edu.au (Q. Yang )

[1]Tel: +61 7 3138 2890; Fax: +61 7 3138 2310.

*2010 MSC:* 35R11

## 1. Introduction

Fractional differential equations have been increasingly used as a powerful tool to model the non-locality and spatial heterogeneity inherent in many real-world problems. As an illustration of this fact, the following books on fractional calculus, anomalous diffusion and its applications have all been published within the last five years: Baleanu et al. [1, 2], Klages et al. [3], Meerschaert and Sikorskii [4], Klafter et al. [5], Mainardi [6], Tarasov [7], Sabatier et al. [8], Ortigueira [9]. For the most recent and up-to-date developments on fractional models across a wide range of disciplines, the interested reader is strongly recommended to consult these excellent works, all by eminent experts in the field.

The booming popularity of fractional models has stimulated demand for efficient solution techniques which can provide rapid insight and visualisation into solution behaviours. It is well-known that analytical solutions are available only for some special, simple (usually linear) fractional models. To solve more general fractional models (either linear or nonlinear), numerical solution techniques are preferred. During the last decade, a large amount of work has been undertaken in this area by many authors, including finite difference methods (e.g. [10, 11, 12, 13]), finite element methods (e.g. [14, 15, 16]), finite volume methods (e.g.[17, 18, 19]), spectral methods (e.g. [20, 21]) and mesh-free methods (e.g. [22, 23]).

A constant challenge faced by researchers in this area is the high computational expense of obtaining numerical solutions to fractional differential equations, owing to the non-local nature of fractional derivatives. The search for high-efficiency numerical methods that can significantly reduce the amount of computational time has become a new trend in the literature.

Preconditioning and Krylov subspace techniques have been a common theme in this context, with authors seeking to reduce the cost of solving the (typically dense) linear systems or matrix function equations that arise from spatial discretisations of fractional differential equations. Yang et al. [15, 18, 24] have developed preconditioners based on eigenvalue deflation. Burrage et al. [16] considered both algebraic multigrid and incomplete LU preconditioning. For the two-sided space-fractional diffusion equation, Moroney and Yang have proposed fast Poisson preconditioner [25] and banded preconditioner [26].

In this paper, we will showcase the high-efficiency of our preconditioned Lanczos method by solving the space-fractional reaction-diffusion equation

$$\frac{\partial u}{\partial t} = -K_\alpha(-\nabla^2)^{\alpha/2}u + g(u), \quad (x, y) \in \Omega,\ t > 0, \tag{1}$$

with homogeneous Neumann boundary conditions on two-dimensional unstructured meshes. The fractional Laplacian operator $-(-\nabla^2)^{\alpha/2}$ of order $1 < \alpha \leq 2$ is defined on the finite domain $\Omega$ through its eigenfunction expansion [27].

The spatial discretisation of (1) is obtained using the matrix transfer technique [27]. With this approach, any standard method such as finite differences, finite elements, finite volumes, etc. may be used to discretise the non-fractional operator $-\nabla^2$, yielding a matrix representation $\mathbf{A}$ of the operator. The discrete representation of the fractional Laplacian $-(-\nabla^2)^{\alpha/2}$ is then simply $-\mathbf{A}^{\alpha/2}$. Under the matrix transfer technique, the semidiscrete form of (1) is thus

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = -K_\alpha\mathbf{A}^{\alpha/2}\mathbf{u} + g(\mathbf{u}), \tag{2}$$

where $\mathbf{u}(t)$ is a vector-valued function approximating $u(x_i, y_i, t)$ at each mesh node $(x_i, y_i)$.

Though the standard spatial discretisations of the Laplacian give rise to a sparse matrix $\mathbf{A}$, the fractional power $\mathbf{A}^{\alpha/2}$ is dense. Hence, methods for solving (2) that avoid forming $\mathbf{A}^{\alpha/2}$ explicitly are preferred. Previously, Yang et al. [15] showed how to use Krylov subspace methods to solve (2) without forming $\mathbf{A}^{\alpha/2}$ when $\mathbf{A}$ is generated using finite differences or finite elements under homogeneous Dirichlet boundary conditions. In the former case, $\mathbf{A}$ is symmetric positive definite, and the standard Lanczos method was used. In the latter case, $\mathbf{A}$ becomes non-symmetric due to the influence of the mass matrix, and the authors used the M-Lanczos method. In both cases, preconditioning was applied to the Krylov subspace method in order to speed convergence, however, this was more challenging in the case of the finite element / M-Lanczos method. In Yang et al. [18], the finite volume method was used to generate $\mathbf{A}$, and it was shown that although the matrix is non-symmetric, a simple transformation allows the standard Lanczos method to be used in this case.

In the present work, we further improve the method by extending the preconditioner first developed by Yang et al. [15] for finite differences to the present method using finite volumes. This new method represents the most versatile numerical scheme of its type for solving (2), being applicable on completely unstructured meshes, while retaining the simplicity and efficiency of the original finite difference-based method.

The outline of the paper is as follows. In Section 2, we propose the numerical scheme for approximating (1) using a finite volume method. With the help of the matrix transfer technique, the solution is written in terms of a matrix-function-vector product $f(\mathbf{A})\mathbf{b}$, where $\mathbf{A}$ is non-symmetric. In Section 3, we investigate the matrix-function approximation techniques for computing $f(\mathbf{A})\mathbf{b}$. In particular, we review the technique for applying the standard Lanczos method to the non-symmetric matrix. In the aspect of accelerating the convergence of the Lanczos method, we highlight the extension of adapting the preconditioner which was previously used for homogeneous Dirichlet conditions to homogeneous Neumann conditions, which gives rise to numerical issues different than Dirichlet conditions as the discrete Laplacian matrix $\mathbf{A}$ is now singular. Section 4 gives detailed numerical results and analysis of the fractional Fisher equation. Section 5 gives some conclusions from this work.

## 2. A finite volume numerical scheme

In this section, we outline the numerical scheme for solving the space-fractional reaction-diffusion equation (1) under homogeneous Neumann boundary conditions. We discretise in space using the vertex-centred finite volume method, which begins with a triangulation of the domain $\Omega$. Let the number of the nodes in the triangulation be denoted $N$. Around each node, we construct a control volume (CV) by connecting element centroids to face midpoints, as described in Ewing et al. [28]. This generates a dual mesh of control volumes, and we denote by $V_i$ the $i$th CV, which has area $\Delta V_i$. These CVs form a partition of the domain $\Omega$, so that $\Omega = \bigcup_{i=1}^{N} V_i$.

According to the matrix transfer technique [27], to derive the finite volume discretisation of (1), one first begins by considering the non-fractional equation

$$\frac{\partial u}{\partial t} = -K_\alpha(-\nabla^2)u + g(u). \tag{3}$$

Integrating (3) over a control volume $V_i$, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V_i} u\,\mathrm{d}V = -K_\alpha \int_{\Gamma_i} (-\nabla u \cdot \mathbf{n})\,\mathrm{d}\sigma + \int_{V_i} g(u)\,\mathrm{d}V \tag{4}$$

where the order of differentiation and integration has been interchanged on the left, and the divergence theorem has been applied to the first term on the right in order to write it as an integral over the CV boundary $\Gamma_i$. Letting $u_i$ denote the numerical solution at the $i$th node, we make the standard approximations $\int_{V_i} u\,\mathrm{d}V = \Delta V_i u_i$ and $\int_{V_i} g(u)\,\mathrm{d}V = \Delta V_i g(u_i)$ and approximate the surface integral by the sum of midpoint rule approximations over each face, to obtain

$$\Delta V_i \frac{\mathrm{d}u_i}{\mathrm{d}t} = -K_\alpha \sum_{j\in F_i} (-\nabla u \cdot \mathbf{n})_{\mathrm{mp}_j} \Delta A_j + \Delta V_i g(u_i), \tag{5}$$

where $F_i$ denotes the set of CV faces comprising $\Gamma_i$, $\Delta A_j$ is the length of the $j$th CV face and $\mathrm{mp}_j$ denotes the midpoint of the $j$th CV face. Interpolation is required to approximate the flux at each CV face midpoint. The underlying triangular mesh provides the means for this, with standard linear shape functions used to compute a constant gradient for each triangle. In this way the total flux across the CV boundary is computed as a linear function of the nodal value $u_i$ and the values $u_j$ for any node $j$ sharing an element with node $i$.

By imposing equation (5) at each mesh node, a system of differential equations is obtained

$$\mathbf{M}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = -K_\alpha \mathbf{K}\mathbf{u} + \mathbf{M}g(\mathbf{u}) \tag{6}$$

where $\mathbf{u} = [u_1, u_2, ..., u_N]^T$ is the numerical solution approximating $u(x_i, y_i, t)$ at each mesh node $(x_i, y_i)$. The matrix $\mathbf{M} = \mathrm{diag}(\Delta V_i)$ is diagonal and represents the contributions from the CV areas. The matrix $\mathbf{K}$ is sparse, symmetric, positive semi-definite, and represents the contributions from each node towards the total flux through each CV. We note that $\mathbf{K}$ possesses a single zero eigenvalue owing to the Neumann boundary conditions imposed on the problem.

By writing (6) as

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = -K_\alpha \mathbf{A}\mathbf{u} + g(\mathbf{u}) \tag{7}$$

and comparing with (3), we identify

$$\mathbf{A} = \mathbf{M}^{-1}\mathbf{K} \tag{8}$$

as the finite volume representation of the negative Laplacian $(-\nabla^2)$.

Having obtained the matrix representation of this operator under the finite volume discretisation, the representation of the fractional Laplacian $-(-\nabla^2)^{\alpha/2}$ is simply $-\mathbf{A}^{\alpha/2}$ using the matrix transfer technique [27]. Hence we derive the spatial discretisation (2) for the space-fractional diffusion equation (1).

We now discretise in time using a mixed implicit-explicit scheme. Let $t_n = n\tau$ for $n = 0, 1, ...$, where $\tau$ is the timestep. Integrating (2) from time $t_n$ to time $t_{n+1}$, we obtain

$$\mathbf{u}(t_{n+1}) - \mathbf{u}(t_n) = -\int_{t_n}^{t_{n+1}} K_\alpha \mathbf{A}^{\alpha/2}\mathbf{u}\,\mathrm{d}t + \int_{t_n}^{t_{n+1}} g(\mathbf{u})\,\mathrm{d}t. \tag{9}$$

Treating the flux term implicitly, and the source term explicitly, both to first order in time, we derive the fully discrete equation

$$\mathbf{u}^{n+1} = (\mathbf{I} + \tau K_\alpha \mathbf{A}^{\alpha/2})^{-1}(\mathbf{u}^n + \tau g(\mathbf{u}^n)) \tag{10}$$

where $\mathbf{u}^n$ denotes the numerical solution vector at time $t_n$. We note that the stability of this first-order temporal scheme is determined by the stability of the explicit Euler method applied to the nonlinear source term $g$ [29].

To advance the solution in time, we write the solution (10) in terms of the matrix-function-vector product

$$\mathbf{u}^{n+1} = f(\mathbf{A})\mathbf{b}^n \tag{11}$$

where $f(\mathbf{A}) = (\mathbf{I}+\tau K_\alpha \mathbf{A}^{\alpha/2})^{-1}$ and $\mathbf{b}^n = \mathbf{u}^n + \tau g(\mathbf{u}^n)$. In the next section we discuss how to use a preconditioned Lanczos method to obtain this matrix-function-vector product (11) without ever forming the dense matrix $\mathbf{A}^{\alpha/2}$.

## 3. Preconditioned Lanczos method

The standard Lanczos approximation to the matrix-function-vector product $f(\mathbf{A})\mathbf{b}$, for symmetric $\mathbf{A}$ is (see, for example, van der Vorst [30]):

$$f(\mathbf{A})\mathbf{b} \approx ||\mathbf{b}||\mathbf{V}_m f(\mathbf{T}_m)\mathbf{e}_1, \quad \mathbf{b} = ||\mathbf{b}||\mathbf{V}_m\mathbf{e}_1 \tag{12}$$

where

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_m\mathbf{T}_m + \beta_m\mathbf{v}_{m+1}\mathbf{e}_m^T \tag{13}$$

is the Lanczos decomposition of $\mathbf{A}$, $\mathbf{T}_m$ is symmetric and tridiagonal, and the columns of $\mathbf{V}_m$ form an orthonormal basis for the Krylov subspace $\mathrm{K}_m(\mathbf{A},\mathbf{b}) = \mathrm{span}\{\mathbf{b},\mathbf{A}\mathbf{b},...,\mathbf{A}^{m-1}\mathbf{b}\}$ with $m \ll n$. The matrix-function-vector product $f(\mathbf{A})\mathbf{b}$ can therefore be approximated by computing the much smaller tridiagonal matrix function $f(\mathbf{T}_m)$ via (12) and (13).

There are several difficulties in applying this method to the finite volume representation of the Laplacian. First, the matrix $\mathbf{A} = \mathbf{M}^{-1}\mathbf{K}$ from equation (8) is not symmetric, despite the fact that the Laplacian itself is a self-adjoint operator. Second, without some form of preconditioning, the rate at which the Lanczos approximation converges can be unacceptably slow.

In our previous work [18], we addressed the first issue, by introducing the matrix $\tilde{\mathbf{A}} = \mathbf{M}^{-1/2}\mathbf{K}\mathbf{M}^{-1/2}$, which is symmetric and similar to $\mathbf{A}$, and showing that

$$f(\mathbf{A}) = \mathbf{M}^{-1/2}f(\tilde{\mathbf{A}})\mathbf{M}^{1/2}. \tag{14}$$

Using this relationship, the matrix-function-vector product (11) can be computed by applying the Lanczos method (12) to the symmetric matrix $\tilde{\mathbf{A}}$ as follows

$$\mathbf{u}^{n+1} = f(\mathbf{A})\mathbf{b}^n = \mathbf{M}^{-1/2}f(\tilde{\mathbf{A}})\tilde{\mathbf{b}}, \quad \tilde{\mathbf{b}} = \mathbf{M}^{1/2}\mathbf{b}^n. \tag{15}$$

Importantly, $\tilde{\mathbf{A}}$ itself need never be formed, since only matrix-vector products are required for the Lanczos method, and these can be computed by observing that $\tilde{\mathbf{A}}\mathbf{v} = \mathbf{M}^{-1/2}(\mathbf{K}(\mathbf{M}^{-1/2}\mathbf{v}))$ for any vector $\mathbf{v}$. Furthermore, since $\mathbf{M}$ is diagonal, the products $\mathbf{M}^{\pm 1/2}\mathbf{v}$ are simply row scalings and attract little cost.

We now address the second issue, of slow convergence, by adapting a preconditioner that has previously been used for finite difference Laplacian matrices [15]. The purpose of this preconditioner is to deflate the smallest $k$ eigenvalues of $\tilde{\mathbf{A}}$ by shifting them to the middle of the spectrum, so that convergence of the Lanczos method proceeds according to the more favourable, modified spectrum.

We suppose that the smallest $k$ eigenvalues $\{\lambda_i\}_{i=1}^k$ and corresponding eigenvectors $\{\mathbf{q}_i\}_{i=1}^k$ of $\tilde{\mathbf{A}}$ have been computed. Then setting $\mathbf{Q}_k = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_k]$ and $\mathbf{\Lambda}_k = \text{diag}\{\lambda_1, ..., \lambda_k\}$, Baglama et al. [31] and Erhel et al. [32] have both proposed the preconditioner $\mathbf{Z}^{-1}$ taking the form

$$\mathbf{Z}^{-1} = \lambda^* \mathbf{Q}_k \mathbf{\Lambda}_k^{-1} \mathbf{Q}_k^T + \mathbf{I} - \mathbf{Q}_k \mathbf{Q}_k^T \tag{16}$$

where $\lambda^* = (\lambda_{\min} + \lambda_{\max})/2$ is the value to which the smallest $k$ eigenvalues will be mapped. The key observation is the following relationship between $f(\tilde{\mathbf{A}})$ and $f(\tilde{\mathbf{A}}\mathbf{Z}^{-1})$ [33]:

$$f(\tilde{\mathbf{A}})\tilde{\mathbf{b}} = \mathbf{Q}_k f(\mathbf{\Lambda}_k) \mathbf{Q}_k^T \tilde{\mathbf{b}} + f(\tilde{\mathbf{A}}\mathbf{Z}^{-1})\hat{\mathbf{b}} \tag{17}$$

where $\hat{\mathbf{b}} = (\mathbf{I} - \mathbf{Q}_k \mathbf{Q}_k^T)\tilde{\mathbf{b}}$. Furthermore, $\tilde{\mathbf{A}}\mathbf{Z}^{-1}$ is symmetric whenever $\tilde{\mathbf{A}}$ is symmetric [33]. Hence, we can apply the Lanczos method to $\tilde{\mathbf{A}}\mathbf{Z}^{-1}$ rather than $\tilde{\mathbf{A}}$.

The goal of preconditioning is to reduce the number of iterations required for the accuracy of the approximation (12) to fall below a given tolerance. Yang et al. [15] have derived the error bound

$$||f(\tilde{\mathbf{A}})\tilde{\mathbf{b}} - ||\tilde{\mathbf{b}}|| \mathbf{V}_m f(\mathbf{T}_m)\mathbf{e}_1|| \leq f(\tilde{\lambda}_1)||\mathbf{r}_m|| \tag{18}$$

for the unpreconditioned problem, where $\mathbf{r}_m = -||\tilde{\mathbf{b}}||\beta_m(\mathbf{e}_m^T \mathbf{T}_m^{-1}\mathbf{e}_1)\mathbf{v}_{m+1}$ is the residual for the Full Orthogonalisation Method (FOM) applied to the linear system $\tilde{\mathbf{A}}\mathbf{x} = \tilde{\mathbf{b}}$ and $\tilde{\lambda}_1$ is the smallest eigenvalue of $\mathbf{T}_m$. Since by (17) the preconditioner introduces no additional error into the approximation, an identical bound holds for the preconditioned problem, with $\tilde{\mathbf{A}}$ replaced by $\tilde{\mathbf{A}}\mathbf{Z}^{-1}$ and $\tilde{\mathbf{b}}$ replaced by $\hat{\mathbf{b}}$ in (18). In our numerical experiments, we will demonstrate the effectiveness of the preconditioner at reducing the number of iterations $m$ required to achieve a given tolerance level.

At this point, we notice that the preconditioner $\mathbf{Z}^{-1}$ originally derived by Baglama et al. [31] and Erhel at al. [32] is under the assumption that $\tilde{\mathbf{A}}$ is nonsingular. Indeed, the very notation $\mathbf{Z}^{-1}$ suggests this. However, in practice $\mathbf{Z}^{-1}$ itself is never formed, since only the matrix-vector products with $\tilde{\mathbf{A}}\mathbf{Z}^{-1}$ are required for the Lanczos method. Yang et al. [24] showed that for any vector $\mathbf{v}$, the product $\tilde{\mathbf{A}}\mathbf{Z}^{-1}\mathbf{v}$ can be computed using the simpler formula

$$\tilde{\mathbf{A}}\mathbf{Z}^{-1}\mathbf{v} = \tilde{\mathbf{A}}\mathbf{v} + \mathbf{Q}_k\mathbf{\Omega}_k\mathbf{Q}_k^T\mathbf{v} \tag{19}$$

where $\mathbf{\Omega}_k = \lambda^*\mathbf{I} - \mathbf{\Lambda}_k$. This formula involves no division by eigenvalues, and hence is applicable even in the case where $\tilde{\mathbf{A}}$ is singular, such as when Neumann boundary conditions are imposed.

One final question is how to efficiently compute the smallest k eigenvalues and eigenvectors of $\tilde{\mathbf{A}}$ required in the preconditioner. A natural approach is to apply the Lanczos method itself [34]. In other words, perform a single initial cycle of unpreconditioned Lanczos iteration, before commencing the time-stepping proper, in order to determine spectral information that will accelerate every subsequent Lanczos iteration. We also note that for the problem with Neumann boundary conditions, the zero eigenvalue and associated eigenvector of ones is known, and can be deflated as part of the initial cycle.

The entire procedure for computing (11) is now summarised.

1. Perform an initial cycle of unpreconditioned Lanczos iteration to determine the $k$ smallest eigenvalues $\{\lambda_i\}_{i=1}^k$ and corresponding eigenvalues $\{\mathbf{q}_i\}_{i=1}^k$ of $\tilde{\mathbf{A}}$.
2. Form the matrices $\mathbf{Q}_k = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_k]$, $\mathbf{\Lambda}_k = \text{diag}\{\lambda_1, ..., \lambda_k\}$ and $\mathbf{\Omega}_k = \lambda^*\mathbf{I} - \mathbf{\Lambda}_k$.
3. For each timestep $n = 0, 1, ...$
   (a) Form the vectors $\mathbf{b}^n = \mathbf{u}^n + \tau g(\mathbf{u}^n)$, $\tilde{\mathbf{b}} = \mathbf{M}^{1/2}\mathbf{b}^n$ and $\hat{\mathbf{b}} = \tilde{\mathbf{b}} - \mathbf{Q}_k(\mathbf{Q}_k^T\tilde{\mathbf{b}})$.
   (b) Perform the preconditioned Lanczos iteration on the vector $\hat{\mathbf{b}}$ to obtain the decomposition $\tilde{\mathbf{A}}\mathbf{Z}^{-1}\mathbf{V}_m = \mathbf{V}_m\mathbf{T}_m + \beta_m\mathbf{v}_{m+1}\mathbf{e}_m^T$, $\hat{\mathbf{b}} = ||\hat{\mathbf{b}}||\mathbf{V}_m\mathbf{e}_1$, stopping the first time that the error bound $f(\tilde{\lambda}_1)||\mathbf{r}_m||$ falls below the prescribed tolerance. The Krylov subspace $\text{K}_m(\tilde{\mathbf{A}}\mathbf{Z}^{-1}, \hat{\mathbf{b}}) = \text{span}\{\hat{\mathbf{b}}, \tilde{\mathbf{A}}\mathbf{Z}^{-1}\hat{\mathbf{b}}, ..., (\tilde{\mathbf{A}}\mathbf{Z}^{-1})^{m-1}\hat{\mathbf{b}}\}$ is formed

by computing matrix-vector products $\tilde{\mathbf{A}}\mathbf{Z}^{-1}\mathbf{v} = \mathbf{M}^{-1/2}(\mathbf{K}(\mathbf{M}^{-1/2}\mathbf{v}))+ \mathbf{Q}_k(\mathbf{\Omega}_k(\mathbf{Q}_k^T\mathbf{v}))$ according to (19).

(c) Compute $\mathbf{w}_1 = ||\hat{\mathbf{b}}||\mathbf{V}_m f(\mathbf{T}_m)\mathbf{e}_1$ which approximates $f(\tilde{\mathbf{A}}\mathbf{Z}^{-1})\hat{\mathbf{b}}$ according to (12).

(d) Compute $\mathbf{w}_2 = \mathbf{Q}_k(f(\mathbf{\Lambda}_k)(\mathbf{Q}_k^T\tilde{\mathbf{b}}))+\mathbf{w}_1$ which approximates $f(\tilde{\mathbf{A}})\tilde{\mathbf{b}}$ according to (17).

(e) Compute $\mathbf{u}^{n+1} = \mathbf{M}^{-1/2}\mathbf{w}_2$ which approximates $f(\mathbf{A})\mathbf{b}^n$ according to (15).

## 4. Numerical Results

### *4.1. Validation*

We consider equation (1) on the domain $\Omega = \{(x, y)|x^2 + y^2 \leq 1\}$ with source term $g(u) = u(1 - u)$, so that we have Fisher's equation on the unit disk. Homogeneous Neumann conditions are imposed on $\partial\Omega$. The initial condition is $u(x, y, 0) = e^{-10(x^2+y^2)}$, the diffusion coefficient is $K_\alpha = 0.05$ and the fractional order is $\alpha = 1.8$.

This problem is radially symmetric, so we may validate our finite volume solution, which is computed on a two-dimensional unstructured triangular mesh in $(x, y)$ coordinates, against a finite difference solution computed on a one-dimensional uniform mesh using polar coordinates $(r, \theta)$. In polar coordinates, the finite difference matrix representation of the Laplacian with homogeneous Neumann conditions is [35]:

$$\mathbf{A} = \begin{bmatrix} 4 & -4 & & & & & \\ -\frac{1}{2} & 2 & -\frac{3}{2} & & & & \\ & & & \cdots & & & \\ & & & & -\frac{2N-5}{2N-4} & 2 & -\frac{2N-3}{2N-4} \\ & & & & & -2 & 2 \end{bmatrix}$$

where the $i$th row corresponds to the node $r_i = (i - 1)\Delta r$ for $i = 1, 2, \ldots N$. We first compute the benchmark solution at time $T = 1$ by applying the iteration (11) with a mesh spacing of $\Delta r = 2^{-13}$ and timestep of $\tau = 2^{-14}$. The matrix function $f(\mathbf{A})$ is computed by diagonalising the $8192 \times 8192$ tridiagonal matrix $\mathbf{A}$.

We then compute the finite volume solution at the same time $T = 1$ using the method described in Sections 2 and 3 on two different triangular meshes with timestep $\tau = 2^{-14}$. The results are exhibited in Table 1. Each row

Table 1: Validation of the finite volume solution against the benchmark finite difference solution. The error reduces at a rate consistent with second order spatial accuracy.

| Nodes | Diameter (D) | Error (E) | D-ratio | E-ratio |
|---|---|---|---|---|
| 6810 | 0.0356 | 4.024E-4 | | |
| 26993 | 0.0185 | 1.078E-4 | 1.924 | $3.733 \approx 1.924^2$ |

lists the number of nodes in the mesh, the maximum element diameter, the maximum difference (error) between the benchmark finite difference solution (interpolated onto the triangular mesh) and the finite volume solution, and finally the ratios of the element diameters and errors. We observe that the error reduces from the first mesh to the second in a manner consistent with second order accuracy in space, which is the spatial accuracy of the finite volume discretisation of the Laplacian [28].

The small timestep size $\tau = 2^{-14}$ was chosen in order to emphasise the spatial error in this example. However, as we show in our next example, the first order implicit/explicit temporal scheme used for this problem can still deliver acceptable solutions with moderate stepsizes.

*4.2. Comparison and efficiency*

We now consider a problem without radial symmetry, for which a one-dimensional finite difference solution is unavailable. The problem is again Fisher's equation on the unit disk with homogeneous Neumann conditions. We use a triangulation consisting of 1513 nodes and 2900 elements, and place a small initial concentration of 0.01 at the node closest to $(0.1, -0.1)$. We consider two cases: standard diffusion with $\alpha = 2$ and $\mathrm{K}_\alpha = 16 \times 10^{-4}$ and fractional diffusion with $\alpha = 1.5$ and $\mathrm{K}_\alpha = 5 \times 10^{-4}$. We note that whatever the values of $K_\alpha$, the fractional diffusion problem will always reach its carrying capacity "sooner" than the standard diffusion problem, as discussed in [36, 18, 16]. The interest in this example is to compare the manner in which the two solutions evolve, and hence the values of the diffusion coefficients $\mathrm{K}_\alpha$ have been chosen so that both solutions reach 80 percent of their carrying capacity at approximately the same time. The simulation is run to time $t = 25$, with a stepsize of $\tau = 0.1$. The mesh and stepsize resolutions lead to numerical solutions for this problem that, while not fully grid-independent, are sufficiently accurate to illustrate the behaviour under discussion, and also to make useful runtime comparisons.

We also note that $\tau = 0.1$ satisfies the stability requirement for the explicit Euler method applied to the Fisher source term $g(u) = u(1-u)$, which linear stability analysis about the steady-state solution $u = 1$ easily shows to be $\tau \leq 2$.

Figures 1 and 2 illustrate the difference in the progressions of the two solutions towards their common steady state $u = 1$. Figure 1 shows that the fractional-order solution ($\alpha = 1.5$) exhibits slower growth at first, but eventually overtakes the integer-order solution ($\alpha = 2$) to reach its carrying capacity sooner. Figure 2 illustrates spatially why this is the case. At early times, as in the top row of the figure, standard diffusion promotes faster growth near the centre, whereas fractional diffusion implies a sharper profile with less spreading of concentration and hence less overall promotion of growth. However, as time proceeds, the heavier tails for the fractional diffusion result in more significant concentrations reaching the extremities of the domain earlier than for standard diffusion. The subsequent promotion of growth in and around the extremities allows the fractional solution to eventually overtake the standard solution. At the point in the simulation illustrated in the bottom row of Figure 2, the fractional solution is exhibiting significant growth throughout the entire domain, while the standard solution still exhibits very little growth at the points furthest from $(0.1, -0.1)$ (that is, at points furthest from the location for the initial seed of concentration).

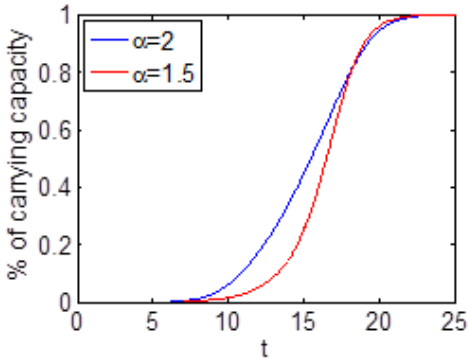


Figure 1: Percentage of carrying capacity versus time for the solution of the (fractional) Fisher's equation on the unit disk. Blue: $\alpha = 2$, $\mathrm{K}_\alpha = 16 \times 10^{-4}$; Red: $\alpha = 1.5$, $\mathrm{K}_\alpha = 5 \times 10^{-4}$

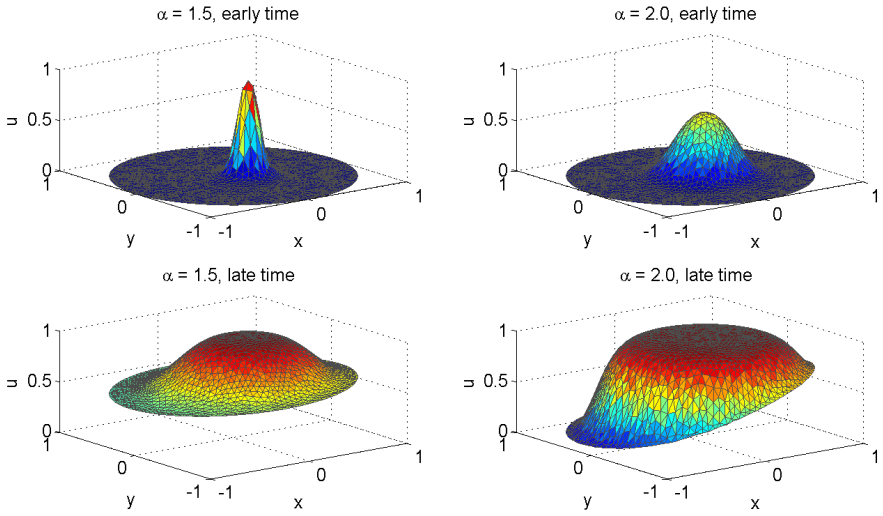


Figure 2: Solutions for the (fractional) Fisher's equation on the unit disk at two points in time. Cases shown are for ($\alpha = 2$, $\mathrm{K}_\alpha = 16 \times 10^{-4}$) and ($\alpha = 1.5, \mathrm{K}_\alpha = 5 \times 10^{-4}$)

We now consider how the preconditioner affects the efficiency with which the results of this simulation can be obtained. We consider the fractional-order case ($\alpha = 1.5$) and plot the number of Lanczos iterations $m$ required for the error bound (18) to drop below $10^{-5}$ at each timestep. Figure 3 plots the values of $m$ against $t$, for various values of $k$ (the number of eigenvalues deflated) given by $k = 1, 2, 5, 10, 20, 50, 100$. Note that there is no reason to consider $k = 0$ , since the smallest eigenvalue is known to be zero *a priori*, and hence can be deflated with no additional work required to estimate its value.

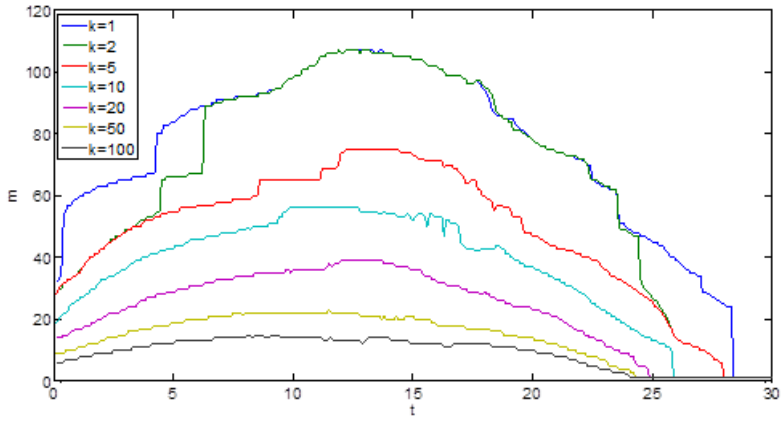


Figure 3: Number of Lanczos iterations required versus time for the fractional Fisher's equation on the unit disk with a 1513 node mesh, $\alpha = 1.5$, $\mathrm{K}_\alpha = 5\times10^{-4}$ , $\tau = 0.1$ and tolerance of $10^{-5}$. Each curve corresponds to a different value of $k$, the number of eigenvalues deflated by preconditioning.

We begin by discussing the general trend of the curves in Figure 3. We see that, whatever the value of $k$, the most Lanczos iterations are required during the middle times of the simulation (around $t = 10$ to $t = 15$). Referring back to Figure 1, we see that this corresponds to the times when the total concentration is changing most rapidly. Hence, it is natural that the largest number of iterations would be required during these times. Towards the end of the simulation, as the solution approaches its steady state, the number of iterations required falls to just one. Again this is consistent with expectations, since the solution is hardly changing at this point. We conclude that using the error bound (18) to determine convergence of the Lanczos iteration performs consistently with expectations for this problem.

Examining the curves in Figure 3 for small $k$ ($k = 1, k = 2$), we see that more than 100 Lanczos iterations are required to compute an acceptable matrix-function-vector product during the middle times of the simulation. With $k = 5$, this number is reduced to fewer than 80 and with $k = 10$ it is fewer than 60. With $k = 50$, the maximum number of iterations required at any time is just 23, and with $k = 100$, it is just 15. Hence, we observe the considerable improvement in efficiency that is provided by the preconditioner.

Figure 4 plots the total runtime for this problem, using MATLAB on a standard desktop PC, against $k$. The figure again confirms the efficiency

gained by preconditioning. With $k = 1$ the runtime is 17.3 seconds, while for $k = 50$ it is only 1.1 seconds, and for $k = 100$ only 0.89 seconds. In fact, from the figure it is clear that even small values of $k$ offer considerable runtime improvement, and that the increase in improvement diminishes as $k$ gets larger. For this problem, values $k \geq 20$ give excellent speed-up.

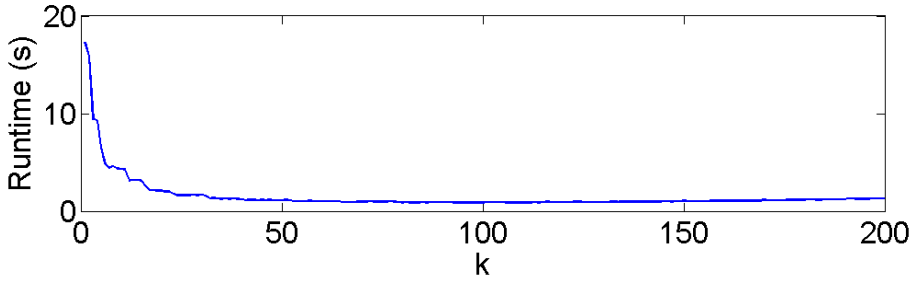


Figure 4: Runtime versus $k$ for the fractional Fisher's equation on the unit disk with a 1513 node mesh, $\alpha = 1.5$, $\mathrm{K}_\alpha = 5 \times 10^{-4}$ , $\tau = 0.1$ and tolerance of $10^{-5}$.

## 5. Conclusions

In this paper we have presented a preconditioned Lanczos method for space-fractional reaction-diffusion equations. The method uses a finite volume spatial discretisation, meaning it is applicable on unstructured meshes. The solution at each timestep is written in terms of a matrix-function-vector product, which is computed iteratively, avoiding the need to form any large, dense matrices. The use of preconditioning is shown to significantly increase the efficiency of the method; an order of magnitude speedup was observed for the particular test problem of the fractional Fisher's equation on a unit disk.

## References

[1] D. Baleanu, Z. B. Guvenc, J. A. T. Machado, New Trends in Nanotechnology and Fractional Calculus Applications, Springer, 2010.

[2] D. Baleanu, J. T. Machado, A. Luo, Fractional Dynamics and Control, Springer, 2012.

[3] R. Klages, G. Radons, I. M. Sokolov, Anomalous Transport: Foundations and Applications, Wiley, 2008.

[4] M. M. Meerschaert, A. Sikorskii, Stochastic Models for Fractional Calculus, De Gruyter, 2011.

[5] J. Klafter, S. C. Lim, R. Metzler, Fractional Dynamics: Recent Advances, World Scientific, 2012.

[6] F. Mainardi, Fractional Calculus and Waves in Linear Viscoelasticity, World Scientific, 2010.

[7] V. E. Tarasov, Fractional Dynamics, Springer, 2011.

[8] J. Sabatier, O. P. Agrawal, J. A. T. Machado, Advances in Fractional Calculus, Springer, 2010.

[9] M. D. Ortigueira, Fractional Calculus for Scientists and Engineers, Springer, 2011.

[10] M. M. Meerschaert, C. Tadjeran, Finite Difference Approximations for Fractional Advection-Dispersion Flow Equations, Journal of Computational and Applied Mathematics 172 (2004) 65–77.

[11] M. M. Meerschaert, C. Tadjeran, Finite difference approximations for two-sided space-fractional partial differential equations, Appl. Numer. Math. 56 (2006) 80–90.

[12] F. Liu, V. Anh, I. Turner, Numerical solution of the space fractional FokkerPlanck equation, Journal of Computational and Applied Mathematics 166 (1) (2004) 209 – 219.

[13] Q. Yang, F. Liu, I. Turner, Numerical methods for fractional partial differential equations with Riesz space fractional derivatives, Applied Mathematical Modelling 34 (1) (2010) 200 – 218.

[14] V. J. Ervin, J. P. Roop, Variational formulation for the stationary fractional advection dispersion equation, Numerical Methods for Partial Differential Equations 22 (3) (2006) 558–576, ISSN 1098-2426, doi: 10.1002/num.20112, URL http://dx.doi.org/10.1002/num.20112.

[15] Q. Yang, I. Turner, F. Liu, M. Ilić, Novel numerical methods for solving the time-space fractional diffusion equation in 2D, SIAM Journal on Scientific Computing 33 (3) (2011) 1159 – 1180.

[16] K. Burrage, N. Hale, D. Kay, An Efficient Implicit FEM Scheme for Fractional-in-Space Reaction-Diffusion Equations, SIAM J. Sci. Comput. 34 (4) (2012) A2145–A2172.

[17] X. Zhang, J. W. Crawford, L. K. Deeks, M. I. Stutter, A. G. Bengough, I. M. Young, A mass balance based numerical method for the fractional advection-dispersion equation: Theory and application, Water Resources Research 41 (7) (2005) n/a–n/a, ISSN 1944-7973, doi:10.1029/2004WR003818, URL http://dx.doi.org/10.1029/2004WR003818.

[18] Q. Yang, T. Moroney, K. Burrage, I. Turner, F. Liu, Novel numerical methods for time-space fractional reaction diffusion equations in two dimensions, ANZIAM J. 52 (2011) C395 – C409.

[19] H. Hejazi, T. Moroney, F. Liu, Stability and convergence of a finite volume method for the space fractional advection-dispersion equation, Journal of Computational and Applied Mathematics 255 (2014) 684 – 697.

[20] X. Li, C. Xu, A Space-Time Spectral Method for the Time Fractional Diffusion Equation, SIAM Journal on Numerical Analysis 47 (3) (2009) 2108–2131, doi:10.1137/080718942.

[21] A. Bueno-Orovio, D. Kay, K. Burrage, Fourier spectral methods for fractional-in-space reaction-diffusion equations, Journal of Computational Physics (Submitted) .

[22] Y. T. Gu, P. Zhuang, Q. Liu, An advanced meshless method for time fractional diffusion equation, International Journal of Computational Methods 08 (04) (2011) 653–665, doi:10.1142/S0219876211002745.

[23] Q. Liu, Y. T. Gu, P. Zhuang, F. Liu, Y. Nie, An implicit RBF meshless approach for time fractional diffusion equations, Computational Mechanics 48 (1) (2011) 1–12, doi:10.1137/080718942.

[24] Q. Yang, T. Moroney, F. Liu, I. Turner, Computationally efficient methods for solving time-variable-order time-space fractional reaction-diffusion equation, Proceedings of the 5th IFAC Symposium on Fractional Differentiation and its Applications, 2012 .

[25] T. Moroney, Q. Yang, Efficient solution of two-sided nonlinear space-fractional diffusion equations using fast Poisson preconditioners, Journal of Computational Physics 246 (2013) 304 – 317.

[26] T. Moroney, Q. Yang, A banded preconditioner for the two-sided, nonlinear space-fractional diffusion equation, Computers & Mathematics with Applications 66 (5) (2013) 659 – 667.

[27] M. Ilić, F. Liu, I. Turner, V. Anh, Numerical approximation of a fractional-in-space diffusion equation (II) with nonhomogeneous boundary conditions, Fract. Calc. Appl. Anal. 9 (2006) 333–349.

[28] R. E. Ewing, T. Lin, Y. Lin, On the accuracy of the finite volume element method based on piecewise linear polynomials, SIAM J. Numerical Analysis 39 (06) (2002) 1865–1888.

[29] J. Frank, W. Hundsdorfer, J. Verwer, On the stability of implicit-explicit linear multistep methods, Applied Numerical Mathematics 25 (2) (1997) 193–205.

[30] H. A. van der Vorst, An iterative solution method for solving $f(A)x = b$ using Krylov subspace information obtained for the symmetric positive definite matrix $A$, J. Comput. Appl. Math. 18 (1987) 249–263.

[31] J. Baglama, D. Calvetti, G. Golub, L. Reichel, Adaptively preconditioned GMRES algorithms, SIAM J. Sci. Comput 20 (1998) 243–269.

[32] J. Erhel, K. Burrage, B. Pohl, Restarted GMRES preconditioned by deflation, J. Comput. Appl. Math. 69 (1996) 303–318.

[33] M. Ilić, I. W. Turner, , V. Anh, A Numerical Solution Using an Adaptively Preconditioned Lanczos Method for a Class of Linear Systems Related with the Fractional Poisson Equation, Journal of Applied Mathematics and Stochastic Analysis (2008) Article ID 104525.

[34] Y. Saad, Numerical Methods for Large Eigenvalue Problems, SIAM, 2011.

[35] B. Bradie, A Friendly Introduction to Numerical Methods, Prentice Hall, 2005.

[36] H. Engler, On the speed of spread for fractional reaction-diffusion equations, Int.J. Diff. Eqn. (2010) Article ID 315421.